1. **In Spring boot how to do if you don't want to run the tomcat server while running the regular class with main method?**

   You can also disable the embedded Tomcat server by specifying the property **spring.main.web-application-type=none** in your **application.properties** or **application.yml** file. This configuration will ensure that the Spring Boot application does not start a web server.

```
spring.main.web-application-type=none
```

2. **What are the methods for incorporating caching into a Spring Boot application?**

   Use caching annotations to indicate which methods should be cached. The primary annotations are @Cacheable, @CachePut, and @CacheEvict.

   @Cacheable: Caches the result of a method call.
   @CachePut: Updates the cache with the method result.
   @CacheEvict: Removes an entry from the cache.

```
@Service
public class MyService {

    @Cacheable("myCache")
    public String getData(String param) {
        // Simulate expensive operation
        return "Data for " + param;
    }

    @CachePut(value = "myCache", key = "#param")
    public String updateData(String param, String newData) {
        // Simulate updating data
        return newData;
    }

    @CacheEvict(value = "myCache", key = "#param")
    public void evictData(String param) {
        // Simulate evicting data from cache
    }
}
```

   Enable caching in your Spring Boot application by adding the **@EnableCaching** annotation to a configuration class or the main application

   Spring Boot works with well-known caching providers like Hazelcast, Ehcache, and Redis.

3. **Can we disable the default web server in the Spring Boot application?**

Yes, we can disable the default web server in the Spring Boot application. To do this, we need to set the **server.port** property to "-1" in the application's **application.properties** file.

4. **Can we create a non-web application in Spring Boot?**

   Yes, we can create a non-web application in Spring Boot. Spring Boot is not just for web applications. Using Spring Boot, we can create applications like Microservices, Console applications, and batch applications.

5. **What Do you understand about Spring Data Rest?**

   Spring Data REST is a framework that exposes Spring Data repositories as RESTful web services. It allows us to expose repositories as REST endpoints with minimal configuration by following Spring Data REST Technologies like **Spring Data** and **Spring MVC** .

6. **Why is Spring Data REST not recommended in real-world applications?**

   Performance – Performance may not be optimal for very large-scale applications.
   Versioning – It can be difficult to version the REST APIs exposed by Spring Data REST.
   Relationships – Handling relationships between entities can be tricky with Spring Data REST.
   Filtering – There are limited options for filtering the results returned by the endpoints.

7. **Explain different phases of RAD model?**

   Business Modeling: Based on the flow of information and distribution between various business channels, the product is designed.
   Data Modeling: The information collected from business modeling is refined into a set of data objects that are significant for the business.
   Application Generation: Automated tools are used for the construction of the software, to convert process and data models into prototypes.

8. **Explain Spring Boot Admin?**

   Spring Boot admin is a community project which helps you to manage and monitor your Spring Boot applications.

9. **What do you mean by hot-swapping in Spring Boot?**

   It is a way to reload the changes without restarting Tomcat, or Jetty server. Eclipse and Many other IDEs support bytecode hot swapping. If you make any changes that don't affect the method signature, it should reload without side effect.

10.